

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, BUET

Quiz of CSE 310 (Compiler Sessional), L-3, T-1, January 2019 Term

Full Marks: 50 Time: 50 Minutes Date: Tuesday, September 17, 2019

Name:

Student No:

(For Examiners' Use Only)

Ques. No.	MCQ (1-10)	11	12	13	14	15	16	17	18	Total
Marks										

Instructions

1. There are 10 multiple choice questions and 8 short questions in the script. Make sure all the questions are printed clearly before you start writing.
2. You should answer the MCQs in the table below.

MCQ Answers

Please write down the answers of the multiple choice questions in the tables below.

Ques.	1	2	3	4	5	6	7	8	9	10
Ans.										

MCQ questions. Select the best answer from the available options. Write the answer (a/b/c/d) in the table above.

(10)

1. Select machine independent phase(s) of the compiler.

i. Symbol table creation,

ii. Syntax analysis,

iii. Intermediate code generation,

iv. Lexical analysis.

(a) All of the mentioned,

(b) i. only,

(c) iii. only,

(d) ii., iii., and iv.
2. Which of the following phase(s) is/are more likely to write to a symbol table?

i. Code optimization,

ii. Syntax analysis,

iii. Semantic analysis,

iv. Lexical analysis.

(a) All of the mentioned,

(b) iii. and iv.,

(c) ii., iii. and iv.,

(d) None of the mentioned.
3. What is the function of yylex()?

(a) scanning

(b) parsing

(c) both of these

(d) neither of these

4. During lexical analysis, which of the following was not inserted in the symbol table?
(a) integer literals **(b) keywords**
(c) character literals (d) operators
5. In bison files, YYSTYPE indicates,
(a) String of token **(b) Data type of token**
(c) Double value (d) None of the above
6. Consider the following rule in a bison file,
`c : a '+' b { $$ = $1 + $2; }`
Now choose what happens when this rule is matched:
(a) a and b are added and saved in c (b) b and c are added and saved in a
(c) a and c are added and saved in b **(d) None of the above**
7. You have used the command “`bison -d -y -v your_roll.y`” to run your .y file. Which part of this command is responsible for generating the y.tab.h file?
(a) **-d** **(b) -y**
(c) -v (d) It is generated by default
8. You have defined many tokens in your .y file. Now, you are wondering what the actual values of these tokens are. Which file will you open to see the values?
(a) y.tab.h (b) y.tab.c
(c) y.output (d) a.out
9. Which of the following assembly code fragment represents the correct translation of the statement `a[1]=10` where a is an array of integers; assume integers are two bytes long and the target machine is byte addressable.
- | | |
|--|--|
| (a) MOV t2,10
MOV t1,1
MOV BX,t1
ADD BX,BX
MOV AX,t2
MOV a[BX],AX | (b) MOV t2,10
MOV t1,1
MOV BX,t1
MOV AX,t2
MOV a[BX],AX |
| (c) MOV t2,10
MOV t1,1
MOV BX,t1
ADD BX,BX
MOV AX,t2
MOV AX,a[BX] | (d) MOV t1,10
MOV t2,1
MOV BX,t1
ADD BX,BX
MOV AX,t2
MOV a[BX],AX |

10. Consider the following lines from the declaration section of a YACC (.y) file.

`%token T_A`

`%type T_B`

`%left OP_A`

`%right OP_B`

Now which of the following is false?

- (a) OP_B is a right associative operator (b) OP_A has lower precedence than OP_B
(c) T_B is a terminal symbol (d) None of the above

Short Questions

11. What are the items stored in symbol table? Which information is used by compiler from symbol table during code generation? (5)
12. Did you manage scopes in your laboratory implementation of symbol table? How scopes are generally managed in a symbol table? (5)

13. Write a flex program which matches a password string containing any number of alphanumeric characters or symbols. However, it must have at least one capital letter, at least one numeric character, and at least one of the following symbols: [~! @ # \$ % ^] anywhere in the string. **(5)**
14. Consider that, you have a lex file `assignment.l` which you have to run using flex in Linux. **(5)** The program requires an input file, which has to be provided as a command line argument. Write the necessary commands to do so.

15. Consider the following grammar rules portion of a yacc code. Assume that the lex file is done accordingly. Now identify how many conflicts this code will generate while compiling with bison. Update this yacc code so that all the conflicts are resolved. (5)

```
%token PROC ID INTEGER REAL START END
%token COLON SEMICOLON ASSIGNOP
%%
P      : D      {printf("One\n");};
D      : ID COLON T      {printf("Two\n");};
D      : PROC ID      {printf("Three\n");}
      | START S END      {printf("Four\n");};
D      : D SEMICOLON D      {printf("Five\n");}
      | ;
T      : INTEGER      {printf("Six\n");}
      | REAL      {printf("Seven\n");};
S      : ID ASSIGNOP ID ;
S      : S SEMICOLON S
      | ;
%%
```

16. Now consider the program below. For this input program, what will be the output of the parser generated from your updated yacc code in the previous question? (5)

```
a: int;
b: real;
proc abc;
  x: int;
  y: real;
start
  x = y;
  a = b;
end
```

17. Write down appropriate code to generate intermediate assembly code for the following rule. The rule represents the while loop structure in C programs. Assume that intermediate codes for all the non-terminals have been appropriately generated from other rules. Also assume that `YYSTYPE` is defined as a pointer to `SymbolInfo` object that contains all the fields you may require. Also you have `newTemp()` and `newLabel()` functions which return a new temporary variable and a new label, respectively. Make any other appropriate assumptions as required. You do not need to perform any error checking. Symbols have their usual meanings. (5)

statement : WHILE LPAREN expression RPAREN statement

18. Consider the block of assembly code below. For convenience the code has been spread into three columns; read the code in the order from top to bottom and left to right. Now, perform as many optimizations as possible on the code. Strike out the lines (i.e., assembly statements) that can be eliminated. Assume all the non-temporary variables (variable names not starting with *ts*) are live at the end of the block. What is the minimum number of temporary variables that should be sufficient for the block? (5)

MOV AX, b2	MOV t4, AX	MOV a2, AX
ADD AX, c2	MOV AX, t4	MOV AX, a2
MOV t0, AX	MOV c2, AX	MOV a2, AX
MOV AX, t0	MOV AX, 0	MOV AX, a2
MOV a2, AX	MOV t5, AX	NEG AX
MOV AX, 1	MOV AX, a2	MOV a2, AX
MOV t1, AX	ADD AX, t5	MOV AX, a2
MOV AX, a2	MOV t6, AX	MOV a2, AX
ADD AX, t1	MOV AX, t6	JMP L1
MOV t2, AX	MOV a2, AX	L0:
MOV AX, t2	MOV AX, 1	MOV AX, b2
MOV c2, AX	MOV t7, AX	NEG AX
MOV AX, 1	MOV AX, t7	MOV b2, AX
MOV t3, AX	CMP AX, 0	MOV AX, b2
MOV AX, c2	JE L0	MOV b2, AX
MOV BX, t3	MOV AX, a2	L1:
MUL BX	NEG AX	MOV AX, 0